REMARKS

Claims 1, 3, 5-9, 11-16, 18-21, and 23-31 are pending. Claims 1, 9, 16, and 21 are independent.

Applicant thanks the Examiner for returning initialed copies of all PTO Form 1449's.

Rejections under 35 U.S.C. § 102(b)

Claim 9

In the action mailed October 21, 2005, claim 9 was rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,530,804 to Edgington et al. (hereinafter "Edgington").

Claim 9 relates to a method of providing instructions to a processor. The method includes loading a plurality of instructions into an emulation instruction register from a test interface, receiving a run-test idle state signal, providing the plurality of instructions to the processor in response to the receipt of the run-test idle state signal, and processing the plurality of instructions without receiving another run-test idle state signal. The run-test idle state signal indicates entry of the test interface into a run-test idle state.

The rejection contends that Edgington describes the receipt of a run-test idle state signal that indicates entry of the test interface into a run-test idle state, where a plurality of instructions are loaded from the test interface into an

emulation instruction register, as recited in claim 9. Applicant respectfully disagrees.

In particular, Edgington includes a *single interface* over which both test and operational instructions are loaded into processor 10. *See, e.g., Edgington*, col. 2, line 56-61. Edgington's test controller operates in conjunction with this "normal" bus interface to exchange data, address, and control information with processor 10 as in normal (non-test) operations so as to not increase the footprint and cost of test equipment. *See, e.g., Edgington*, col. 4, line 53-59; col. 1, line 17-27.

Edgington neither describes nor suggests that this single interface enters into a run-test idle state and that a signal indicative thereof is received. Instead, test controller 21 receives a generate debug mode interrupt (GDMI) signal that indicates that testing is to begin. *See, e.g, Edgington*, col. 11, line 5-19 and col. 3, line 17-20. This GDMI signal has nothing to do with Edgington's "normal" bus interface being in a run test idle state. Rather, Edgington describes that test controller 21 must wait until an interruptible point in the instruction execution stream must be reached before the instruction execution stream can be interrupted and testing can begin. *See, e.g., Edgington*, FIG. 9, elements 72, 74; col. 11, line 20-26; col. 12, line 18-38.

Since Edgington's "normal" bus interface is performing normal operations when the GDMI signal is received, the GDMI signal does not indicate that the "normal" bus interface has entered into a run test idle state. Instead, the GDMI signal *triggers* the interruption of the instruction execution stream.

Therefore, Edgington neither describes nor suggests the receipt of a run-test idle state signal as recited in claim 9. Accordingly, anticipation has not been established. Applicant requests that the rejections of claim 9 and the claims dependent therefrom be withdrawn.

Claim 16

Claim 16 was rejected under 35 U.S.C. § 102(b) as anticipated by Edgington.

Claim 16 relates to a processor that includes a test interface, an emulation instruction register adapted to store a plurality of emulation instructions received from the test interface, emulation control logic adapted to supply the plurality of emulation instructions to a processor pipeline in response to detection of an entry of the test interface into run-test idle state, and a decoder to receive the plurality of instructions for processing.

Edgington's test controller does not detect an entry of the test interface into run-test idle state. As discussed above, Edgington's test controller *interrupts* the instruction execution

stream upon receipt of a GDMI signal. In doing so, Edgington's test controller detects an interruptible point in the execution stream. However, an interruptible point in an execution stream is not a run test idle state of a test interface. Rather, an interruptible point in an execution stream is a point in a stream of instructions.

Therefore, Edgington neither describes nor suggests emulation control logic adapted to supply a plurality of emulation instructions to a processor pipeline in response to detection of an entry of the test interface into run-test idle state, as recited in claim 16. Accordingly, anticipation has not been established. Applicant requests that the rejections of claim 16 and the claims dependent therefrom be withdrawn.

Claim 21

Claim 21 was rejected under 35 U.S.C. § 102(b) as anticipated by Edgington.

Claim 21 relates to an apparatus that includes operating instructions residing on a machine-readable storage medium. The operating instructions are for use in a device to handle a plurality of emulation instructions. The operating instructions cause the device to load the plurality of emulation instructions into a single emulation instruction register, have a test interface enter a run-test idle state, provide the plurality of emulation instructions to a processor in response to entry of

the test interface into the run-test idle state, and process the plurality of emulation instructions.

Edgington's test controller does not have a test interface enter a run-test idle state, nor does it provide a plurality of emulation instructions to a processor in response to entry of a test interface into the run-test idle state, as recited in claim 21. As discussed above, Edgington's test controller *interrupts* the normal instruction execution stream upon receipt of a GDMI signal. The interruption of a normal instruction stream is not an entry into a run-test idle state.

Further, Edgington's test controller provides instructions when it detects an interruptible point in the execution stream. However, an interruptible point in an execution stream is not an entry of a test interface into the run-test idle state. Rather, an interruptible point in an execution stream is a point in a stream of instructions.

Therefore, Edgington neither describes nor suggests having a test interface enter a run-test idle state and providing a plurality of emulation instructions to a processor in response to entry of a test interface into the run-test idle state, as recited in claim 21. Accordingly, anticipation has not been established. Applicant requests that the rejections of claim 21 and the claims dependent therefrom be withdrawn.

Rejections under 35 U.S.C. § 103(a)

Claim 1

Claim 1 was rejected under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 5,774,737 to Nakano (hereinafter "Nakano") and Edgington.

Claim 1 relates to a method that includes receiving a plurality of instructions from a test interface, loading the plurality of instructions into an emulation instruction register, receiving a plurality of instructions from the emulation instruction register, determining a validity of a first instruction of the plurality of instructions by reading width bits in the first emulation instruction, providing the first instruction to a decoder of the processor if the first instruction is valid, without receiving a run-test idle state signal, determining a validity of a second instruction of the plurality of instructions by reading width bits in the second instruction, and providing the second instruction to the decoder if the second instruction is valid. Read width bits define the validity and size of the first and second emulation instructions.

The rejection of claim 1 contends that Nakano describes determining a validity of an instruction by reading width bits in the instruction, wherein the width bits define the validity and size of the instruction. In particular, the rejection is

understood to contend that certain of Nakano's VLIW instructions (i.e., a VLIW-instruction word length rewrite instruction) are such width bits.

Applicant respectfully disagrees. As discussed in Nakano, a VLIW-instruction word length rewrite instruction causes the instruction word length stored in a word length register 11 to be rewritten. *See, e.g., Nakano*, col. 2, line 25-45. This rewritten instruction word length allows *subsequent* VLIW-instruction words to be handled. *See, e.g., Nakano*, col. 9, line 18-25.

Nakano's instruction word length rewrite instructions thus do not define the validity and the size of the instruction in which they are found, as required of the width bits in claim 1. Instead, Nakano's instruction word length rewrite instructions allow subsequent instructions to be handled using the rewritten word length.

Edgington does nothing to remedy this deficiency of Nakano. In particular, Edgington neither describes nor suggests such width bits.

Since elements and/or limitations of claim 1 are neither described nor suggested by Nakano and Edgington, a *prima facie* case of obviousness has not been established. Applicant therefore requests that the rejections of claim 1 and the claims dependent therefrom be withdrawn.

Attorney's Docket No. Intel Corporation:10559-286001/P9293-ADIAPD1872-1-US

Applicant asks that all claims be allowed. No fees are believed due at this time. Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

BY
JOHN F. CONROY
REG. NO. 45,485

Date: January 17, 2006

Scott C. Harris
Reg. No. 32,030

Fish & Richardson P.C.
PTO Customer No. 20985
Attorneys for Intel Corporation
12390 El Camino Real
San Diego, California 92130
(858) 678-5070 telephone
(858) 678-5099 facsimile

10577847.doc